Viewport-Adaptive Navigable 360-Degree Video Delivery

Xavier Corbillon, Gwendal Simon, Alisa Devlic, Jacob Chakareski

Viewport-Adaptive Navigable 360-Degree Video Delivery

Xavier Corbillon Gwendal Simon
IMT Atlantique, France
firstname.lastname@imt-atlantique.fr

Alisa Devlic
Huawei Technologies, Sweden
alisa.devlic@huawei.com

Jacob Chakareski
University Alabama, USA
jacob@ua.edu

***Abstract*—The delivery and display of 360-degree videos on Head-Mounted Displays (HMDs) presents many technical challenges. 360-degree videos are ultra high resolution spherical videos, which contain an omnidirectional view of the scene. However only a portion of this scene is displayed on the HMD. Moreover, HMD need to respond in 10 ms to head movements, which prevents the server to send only the displayed video part based on client feedback. To reduce the bandwidth waste, while still providing an immersive experience, a viewport-adaptive 360-degree video streaming system is proposed. The server prepares multiple video representations, which differ not only by their bit-rate, but also by the qualities of different scene regions. The client chooses a representation for the next segment such that its bit-rate fits the available throughput and a full quality region matches its viewing. We investigate the impact of various spherical-to-plane projections and quality arrangements on the video quality displayed to the user, showing that the cube map layout offers the best quality for the given bit-rate budget. An evaluation with a dataset of users navigating 360-degree videos demonstrates that segments need to be short enough to enable frequent view switches.**

I. Introduction

The popularity of navigable 360-degree video systems has grown with the advent of omnidirectional capturing systems and interactive displaying systems, like Head-Mounted Displays (HMDs). However, to deliver 360-degree video content on the Internet, the content providers have to deal with a problem of bandwidth waste: What is displayed on the device, which is indifferently called *Field of View (FoV)* or *viewport*, is only a fraction of what is downloaded, which is an omnidirectional view of the scene. This bandwidth waste is the price to pay for interactivity. To prevent *simulator sickness* [12] and to provide good Quality of Experience (QoE), the vendors of HMDs recommend that the enabling multimedia systems react to head movements as fast as the HMD refresh rate. Since the refresh rate of state-of-the-art HMDs is 120 Hz, the whole system should react in less than 10 ms. This delay constraint prevents the implementation of traditional delivery architectures where the client notifies a server about changes and awaits for the reception of content adjusted at the server. Instead, in the current Virtual Reality (VR) video delivery systems, the server sends the full 360-degree stream, from which the HMD extracts the viewport in real time, according to the user head movements. Therefore, the majority of the delivered video stream data are not used.

Let us provide some numbers to illustrate this problem. The viewport is defined by a device-specific viewing angle (typically 120 degrees), which delimits horizontally the scene from the head direction center, called viewport center. To ensure a good immersion, the pixel resolution of the displayed viewport is high, typically 4K (3840×2160). So the resolution of the full 360-degree video is at least 12K (11520×6480). In addition, the immersion requires a video frame rate on the order of the HMD refresh rate, so typically around 100 frames per second (fps). Overall, high-quality 360-degree videos combine both a very large resolution (up to 12K) and a very high frame rate (up to 100 fps). To compare, the bit-rate of 8K videos at 60 fps encoded using High Efficiency Video Coding (HEVC) is around 100 Mbps [16].

We propose in this paper a solution where, following the same principles as in rate-adaptive streaming technologies, the server offers multiple *representations* of the same 360-degree video. But instead of offering representations that only differ by their bit-rate, the server offers here representations that differ by having a Quality Emphasized Region (QER): a region of the video with a better quality than the remaining of the video. Our proposal is a *viewport-adaptive streaming system* and is depicted in Figure 1. The QER of each video representation is characterized by a *Quality Emphasis Center (QEC)*, which is the center of the QER and represents a given viewing position in the spherical video. Around the QEC, the quality of the video is maximum, while it is lower for video parts that are far from the QEC. Similarly as in Dynamic Adaptive Streaming over HTTP (DASH), the video is cut into segments and the client periodically runs an *adaptive algorithm* to select a representation for the next segment. In a viewport-adaptive system, clients select the representation such that the bit-rate fits their receiving bandwidth and the QEC is closest to their viewport center.

This viewport-adaptive 360-degree streaming system has three advantages: (i) the bit-rate of the delivered video is lower than the original full-quality video because video parts distant from the QEC are encoded at low quality. (ii) When the end-user does not move, the viewport is extracted from the highest quality part of the spherical video. And (iii) when the head of the end-user moves, the device can still extract a viewport because it has the full spherical video. If the new

The work of J. Chakareski was supported by NSF award CCF-1528030

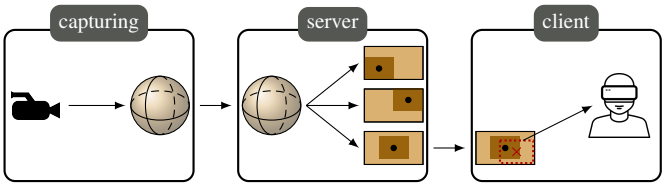

Figure 1: Viewport-adaptive 360-degree video delivery system: The server offers video representations for three QERs. The dark brown is the part of the video encoded at high quality, the light brown the low quality. The viewport is the dotted red rectangle, the viewport center the cross

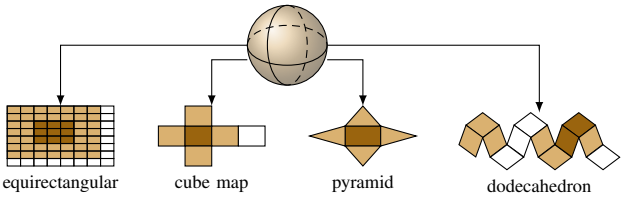

Figure 2: Projections into four geometric layouts

viewport center is far from the QEC of the received video representation, the quality of the extracted viewport is lower but this degradation holds only until the selection of another representation with a closer QEC.

The remainder of the paper is organized as follows. First, we present our viewport-adaptive streaming system, and we show how it can be integrated into the MPEG DASH-VR standard. Our proposal is thus a contribution to the VR group that MPEG launched in May 2016 [13]. Second, we address the choice of the geometric layout into which the spherical video is projected for encoding. We evaluate several video quality arrangements for a given geometric layout and show that the cube map layout with full quality around the QEC and 25 % of this quality in the remaining faces offers the best quality of the extracted viewport. Third, we study the required video segment length for viewport-adaptive streaming. Based on a dataset of real users navigating 360-degree videos, we show that head movements occur over short time periods, hence the streaming video segments have to be short enough to enable frequent QER switches. Fourth, we examine the impact of the number of QERs on the viewport quality and we show that a small number of (spatially-distributed over the sphere) QERs suffices to get high viewport quality. Finally, we introduce a tool (released as open source), which creates video representations for the proposed viewport-adaptive streaming system. The tool is highly configurable: from a given 360-degree video, it allows any arrangement of video quality for a given geometric layout, and it extracts the viewport from any viewport center. This tool thus provides the main software module for the implementation of viewport-adaptive streaming of navigable 360-degree videos.

II. Background and Related Work

We introduce the necessary geometric concepts for spherical videos, and discuss prospective architecture proposals for navigable 360-degree video delivery.

A. Geometric Layouts for 360-degree Videos

A 360-degree video is captured in every direction from a unique point, so it is essentially a *spherical* video. Since current video encoders operate on a two-dimensional rectangular image, a key step of the encoding chain is to project the spherical video onto a planar surface. The projection of a sphere onto a plane (known as mapping) has been studied for centuries. In this paper, we consider the four projections that are the most discussed for 360-degree video encoding [25]. These layouts are depicted in Figure 2. From the images that are projected on an *equirectangular* panorama, a *cube map*, and a *rhombic dodecahedron*, it is possible to generate a viewport for any position and angle in the sphere without any information loss [2, 14]. However, some pixels are over-sampled (a pixel on the sphere is projected to a pair of pixels in the projected image). This is typically the case for the sphere pole when projected on the equirectangular panorama. This over-sampling degrades the performance of traditional video encoders [25]. On the contrary, the projection into a pyramid layout causes under-sampling: some pairs of pixels on the sphere are merged into a single pixel in the projected image by interpolating their color values. This under-sampling cause distortion and information loss in some extracted viewports. Previous work regarding projection of spherical videos into different geometric layouts focuses on enabling efficient implementation of signal processing functions [7] and improving the video encoding [21].

Our contributions. We propose to leverage the geometric structure of the layouts to implement a video encoding based on QER. Each geometric layout is characterized by a number of *faces* (*e.g.*, 6 for the cube map, 12 for the dodecahedron) and a given *central point* (which corresponds to a position on the sphere). From the given central point and layout, our idea is to encode the front face in full quality while the quality of other faces is reduced. To our knowledge, such idea has not been studied yet. Another originality of our work is that we measure QoE by measuring the quality of several extracted viewports instead of the full projected video.

B. Personalized Viewport-Only Streaming

An intuitive idea to address the problem of resource waste due to the delivery of non-displayed video data is to stream only the part of the video that corresponds to the viewport. This solution however does not enable fast navigation within the 360-degree video: When the client moves the head, the viewport center changes, requiring a new viewport to be immediately displayed. Since the device has no knowledge about other parts of the spherical video, it has to notify the server about the head movement and wait for the reception of the newly adjusted viewport. As seen in other interactive multimedia systems [1], this solution cannot meet the 10 ms latency requirement in the standard Internet, even with the

assistance of content delivery network (CDN). In addition, this solution requires the server to extract a part of the video (thus to spend computing resources) for each client connection.

Our contributions. In our system, the server always delivers the full video, but it has different versions of this video depending on the QER (characterized by its QEC). The client device selects the right representation and extracts the viewport. The storage requirements at the server side increase but all the processing is done at the client side (representation selection and viewport extraction). This idea matches the adaptive delivery solutions that content providers have recently adopted (*e.g.* DASH), trading client-personalized delivery for simple server-side management operation.

C. Tiling for Adaptive Video Streaming

To deal with the cases of end-users consuming only a fraction of the video (navigable panorama [3, 19, 23] and large-resolution video [9]), the most studied delivery solution leverages the concept of *tiling*. The idea is to spatially cut a video into independent tiles. The server offers multiple video representations of each tile; the client periodically selects a representation for each tile and it has to reconstruct the full video from these tiles before the viewport extraction. In a short paper, Ochi et al. [15] have sketched a tile-based streaming system for 360-degree videos. In their proposal, the spherical video is mapped onto an *equirectangular* video, which is cut into 8×8 *tiles*. More recently, Hosseini and Swaminathan [5] proposed a *hexaface sphere*-based tiling of a 360-degree video to take into account projection distortion. They also present an approach to describe the tiles with MPEG DASH Spatial Relationship Description (SRD) formatting principles. Quan et al. [17] also propose the delivery of tiles based on a prediction of the head movements. Zare et al. [26] evaluate the impact of different tiling scheme on the compression efficiency and on the transmission bit-rate saving.

A tile-based adaptive streaming system provides the same features as our proposed system regarding navigability (the clients get the full video), bandwidth waste reduction (the video at low quality for non-viewport part) and QoE maintenance (the downloaded video is at full quality near the viewport center). It has however several critical weaknesses. First, the client has to first reconstruct the video from independent tiles before the viewport extraction can take place, which requires energy and time spent for each video frame. Second, the more tiles there are, the less efficient the video encoding is due to the tile independence [19]. Third, the management at the server is heavier because the number of files is larger. For example, a typical 8×8 tiling offered at six quality levels contributes to having 384 independent files for each video segment, and this results in larger Media Presentation Description (MPD) files (or manifest files). Finally, the management at the client side is heavier. For each tile, the client should run a representation selection process and manage a specific network connection with the server.

Our contributions. In our system, the server prepares n QER-based videos, each of them being a pre-processed set of tile representations. Each QER-based video is then encoded at k *global* quality levels. The main advantages include an easier management for the server (fewer files hence a smaller MPD file), a simpler selection process for the client (by a distance computation), and no need for re-constructing the video before the viewport extraction.

D. QER-Based Streaming

A 360-degree video provider (Facebook) has recently released detailed the implementation of its delivery platform [8]. The spherical video is projected onto a pyramid layout from up to 30 central points to generate a set of video representations. Since the front face of pyramid projection has a better image quality than the other faces, the system is in essence similar to our concept of QER. The end-users periodically select one of the representations based on their viewport center. This implementation corroborates that, from an industrial perspective, the extra-cost of generating and storing multiple QER-based representations of the same video is compensated by bandwidth savings and enhanced system usability. However, as seen in Section IV, the pyramid projection is not the best regarding the viewport quality. Moreover, the system uses the same video quality on each face, which is less efficient than our proposal. Finally, the impact of the video encoding on the solution is not given.

Lee et al. [10] studied in another context the coding of a regular video with a QER. The QER is generated near the area that is the most likely to attract gazes. They do not propose to generate different representations with different QERs.

Our contributions. Our approach is based on the same idea of offering multiple QER-based video representations. However, we provide a complete study of our system with the additional distinction of having varying quality across the geometrical layout. Moreover, our study includes an evaluation of several geometric layouts, an analysis of the best segment duration, an analysis of the best number of QERs, and a step towards integration into MPEG DASH.

III. System architecture

This section describes the system architecture of the proposed navigable 360-degree video delivery framework.

Server. The server takes as an input a 360-degree video in equirectangular format and transforms each frame into a desired geometrical layout. Then, it creates n different video versions, each with a different QER and encoded in k different bit-rates (see Figure 3). The server splits all such encoded videos into segments, which are classified in $n \times k$ representations (based on their respective bit-rate and QER), enabling clients to regularly switch from one representation to another. The video quality around the QEC is the highest, while the remaining part is encoded at lower quality.

Client. Over time the viewer moves the head and the available bandwidth changes. Current HMDs record changes in head orientation through rotation around three perpendicular axes, denoted by *pitch*, *yaw*, and *roll*. Head movements modify the

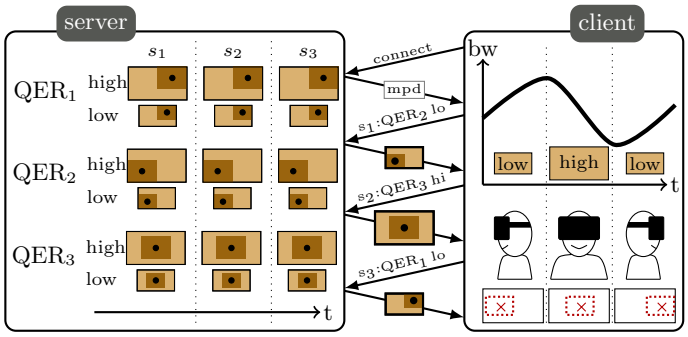

Figure 3: Viewport-adaptive streaming system: the server offers 6 representations (3 QERs at 2 bit-rates). The streaming session lasts for three segments. The client head moves from left to right, the available bandwidth varies. For each segment, the client requests a representation that matches both the viewport and the network throughput.

```
<?xml version="1.0"?>
<MPD>
  <Representation id="1" qec="90,60" bandwidth="9876" width
      ="1920" height="1080" frameRate="30">
    <EssentialProperty schemeIdUri="urn:mpeg:dash:vrd:2017"
        value="0,0">
    <SegmentList timescale="1000" duration="2000">
      ...
  </Representation>
 </AdaptationSet>
</MPD>
</xml>
```

Listing 1: Extensions of MPD file

viewport center, requiring a new viewport to be displayed. State-of-the-art HMDs can perform the extraction [22]. The client periodically sends a request to the server for a new segment in the representation that matches both the new viewport center and the available throughput.

Adaptation algorithm. Similarly to DASH, the client runs an adaptation algorithm to select the video representation. It first selects the QER of the video based on the viewport center and the QECs of the available QERs. This is an important addition to the DASH bit-rate adaptation logic, since the QER determines the quality of the video that is delivered and displayed to the user. After the QER selection, the client chooses the video representation characterized by this QER and whose bit-rate fits with the expected throughput for the next x seconds (*i.e.*, x being the segment length). The server replies with the requested video representation, from which the client extracts the viewport, displaying it on the HMD, as shown in Figure 3.

Rate-adaptive streaming systems are based on the assumption that the selected representation will match the network conditions for the next x seconds. Rate adaptation algorithms are developed [11, 20] to reduce the mismatch between the requested bit-rate and the throughput. In our proposal, the adaptation algorithm should also ensure that the viewport centers will be as close as possible to the QEC of the chosen QER during the x next seconds. In this paper, we implement a simple algorithm for QEC selection: we select the QEC that has the smallest orthodromic distance[1] to the viewport center at the time the client runs the adaptation algorithm. Similarly as for bit-rate adaptation, we expect new viewport-adaptive algorithms to be developed in the future to better predict the head movement and select the QEC accordingly. In their recent paper, Quan et al. [17] have made a first study where they show that a simple linear regression algorithm enables an accurate prediction of head movements for short segment size.

[1]The shortest distance between two points on the surface of a sphere, measured along the surface of the sphere. Its measure is proportional to the radius of the sphere; we refer to "distance unit" to denote the radius size.

Video segment length. A video segment length determines how often requests can be sent to the server. It typically ranges from 1 s to 10 s. Short segments enables quick adaptation to head movement and bandwidth changes, but it increases the overall number of segments and results in larger manifest files. Shorter segments also increase the network overhead due to frequent requests, as well as the network delay because of the round trip time for establishing a TCP connection. Longer segments improve the encoding efficiency and quality relative to shorter ones, however they reduce the flexibility to adapt the video stream to changes. We discuss segment length and head movement in Section IV-B based on a dataset.

Extending the MPD file. To implement the proposed viewport-adaptive video streaming, we extended a DASH MPD file with new information, as illustrated in Listing 1. Each representation contains the `coordinates` of its QEC in degrees, besides the parameters that are already defined in the standard [6]. Those coordinates are the two angles of the spherical coordinates of the QEC, ranging respectively from 0 d to 360 degrees and from −90 d to 90 degrees. All representations from the same adaptation set should have the same reference coordinate system. The `@schemeIdUri` is used to indicate some extra information on the video such as the video source id and the projection type. The projection type is used by the client to determine if he knows how to extract viewports from this layout.

IV. System Settings

The preparation of 360-degree videos for viewport-adaptive streaming relies on multiple parameters. We distinguish between global parameters (the number of QERs, the number of representations, the segment length and the geometric layout) and local (*per representation*) parameters (the target bit-rate, the number of different qualities in a representation, the quality arrangement of different faces of a geometric layout). We will not be comprehensive regarding the selection of all these parameters here. Some of them require a deeper study related to signal processing, while others depend on business considerations and infrastructure investment. In this paper, we restrict our attention to three key questions: What is the best geometric layout to support quality-differentiated 360-

degree video? What is the best segment length to support head movements, while maintaining low management overhead? What is the best number of QERs n to reduce the induced storage requirements, while offering a good QoE? To answer these three questions, we have developed a software tool and used a dataset from a real VR system.

Dataset. We graciously received from Jaunt, Inc a dataset recording the head movements of real users watching 360-degree videos. The dataset is the same as the one used by Yu et al. [25]. It comprises eleven omnidirectional videos that are ten seconds long. These videos are typical of VR systems. The dataset contains the head movements of eleven people who were asked to watch the videos on a state-of-the-art HMD (Occulus Rift DK2). The subjects were standing and they were given the freedom to turn around, so the head movements are of wider importance than if they were asked to watch the video while sitting. Given the length of the video and the experimental conditions, we believe that the head movements thus correspond to a configuration of wide head movements, which is the most challenging case for our viewport-adaptive system. Yu et al. [25] studied the most frequent head positions of users. We are interested here in head movements during the length of a segment.

Software. We have developed our own tool to manipulate the main concepts of viewport-adaptive streaming. Since the code is publicly available,[2] the software can be used to make further studies and to develop real systems. The main features include:

- *Projection from a spherical video onto any of the four geometric layouts and vice versa*. The spherical video is the pivot format from which it is possible to project to any layout. Our tool rotates the video so that the QEC is always at the same position on the 2D layout.
- *Adjusting the video quality for each geometric face of any layout*. For each face, we set the resolution in number of pixels and the target encoding bit-rate.
- *Viewport extraction for any viewport center on the sphere*. It includes the decoding, rescaling and "projection" of each face of the input video to extract the viewport. This tool support extraction of viewport that overlap on multiple faces with different resolution and bit-rate target.

A. Geometric Layout

We report now the experiment of measuring the video quality of viewports, extracted from 360-degree videos projected onto various geometric layouts and with various face quality arrangements. We used two reference videos.

- *The original equirectangular video at full quality*:[3] We extract viewports at 1080p resolution from this 4K equirectangular video, which represents the reference (original) video used to assess the objective video quality.
- *The same equirectangular video re-encoded at a target bit-rate*. It is what a regular delivery system would deliver for the same bit-rate budget (here 6 Mbps being 75 % of the original video bit-rate). We re-encoded the original full-quality video with HEVC by specifying this bit-rate target. We call it *uniEqui* to state that, in this video, the quality is uniform.

The performance of the layout can be studied with regards to two aspects: (i) *the best viewport quality*, which is the quality of the extracted viewport when the viewport center and the QEC perfectly matches, (ii) and the *sensitivity to head movements*, which is the degradation of the viewport quality when the distance between the viewport center and the QEC increases. To examine both aspects, we select one QEC on the spherical video. We chose one orthodromic distance d that will vary from 0 to π. We extract a ten seconds long viewport video, at distance d from the QEC, at the same spherical position on the original equirectangular video and on the tested video. We used two objective video quality metrics to measure the quality of the extracted viewport compared to the original full quality viewport: Multiscale - Structural Similarity (MS-SSIM) [24] and Peak Signal Noise to Ratio (PSNR). MS-SSIM compares image by image the structural similarity between this video and the reference video. The PSNR measures the average error of pixel intensities between this video and the reference video. The MS-SSIM metric is closer to human perception but is less appropriate than the PSNR to measure difference between two measurements. Since we compare several encoded versions of the *same* viewport against the original, these well-known tools provide a fair performance evaluation of viewport distortion. We perform multiple quality assessment (typically forty) at the same distance d but at different positions and average the result.

We represent in Figure 4 the video quality (measured by MS-SSIM) of the viewport that is extracted from our quality-differentiated layouts (equirectangular panorama with 8×8 tiles, cube map, pyramid, and dodecahedron). We also represent by a thin horizontal line the video quality of the same viewports extracted from the *uniEqui* layout (it does not depend on the distance since the quality is uniform). For each geometric layout, we have tested numerous quality arrangements with respect to the overall bit-rate budget. We selected here the "best" arrangement for each layout. For the cube map, the QEC is located at the center of a face. This face is set at full quality (same bit-rate target as the same portion of the original video), and the other faces at 25 % of the full quality target.

The projection on a cube map appears to be the best choice for the VR provider. The quality of the viewport when the QEC and the viewport center matches ($d = 0$) is above 0.98, which corresponds to imperceptible distortion relative to the full quality video. For all layouts, the quality decreases when the distance d increases but the quality for the cube map layout is always the highest. Note that the pyramid projection (the layout chosen by Facebook [8]) is especially sensitive to head movements. The viewport extracted from a cube map projection has a better quality than that extracted from the *uniEqui* for viewport center for up to 2 units from the QEC

[2]https://github.com/xmar/360Transformations

[3]https://youtu.be/yarcdW91djQ

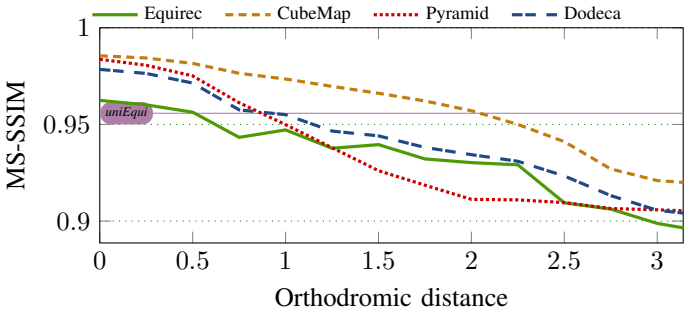

Figure 4: Average MS-SSIM depending on the distance to the QEC for the four geometric layouts. Global bit-rate budget 6 Mbps

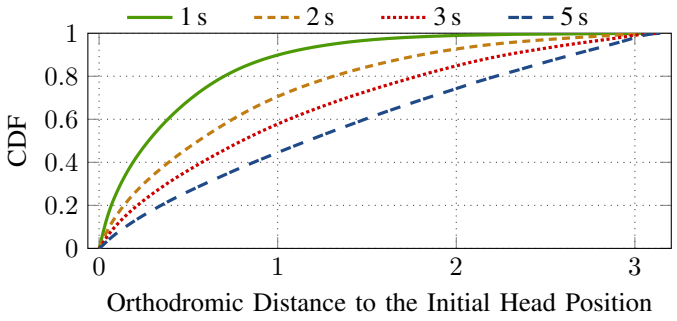

Figure 5: CDF of the time spent at distance d from the head position on the beginning of the segment, for various segment lengths.

while the other layouts viewports increase a video quality for only 1 unit of the QEC. We study next the interplay between this distance, the segment length and the number of QECs.

B. Segment Length

The segment length is a key aspect of viewport-adaptive streaming. Long segments are easier to manage and better for video encoding, but short segments enable fast re-synchronisation to head movement. With respect to Figure 4, the segment length should be chosen such that the distance between the viewport center and the QEC are rarely higher than 1.5 distance units.

Given the dataset, we show the distribution of head movements for various segment lengths in Figure 5. For each video and person watching it, we set timestamps that correspond to the starting time of a video segment, *i.e.,* the time at which the users select a QEC. Then, we measure the orthodromic distance between this initial head position and every viewport center during the next x seconds, where x is the segment length. In Figure 5, we show the cumulative density function (CDF) of the time spent at a distance d from the initial head position. A point at $(1.5, 0.6)$ means that, on average, users spend 60 % of their time with a viewport center at less than 1.5 distance units from the viewport center on the beginning of the segment.

Our main observation is that viewport-adaptive streaming requires short segment lengths, typically smaller than 3 s. Indeed, for a segment length of 5 s, users spend on average half of their time watching at a position that is more than 1.3 distance units away from the initial head position, which results in a degraded video quality. A segment length of 2 s appears to be a good trade-off: 92 % of users never diverged to a head position that is further than 2 distance unit away from the initial head position, and users can experience the full video quality three quarters of the time (head distance lesser than 0.7 distance unit). Please recall that our dataset captures a challenging experiment for our system. We can expect narrower head movements, and thus longer possible segment lengths, for sitting users and longer videos. Note also that these results are consistent with the head movement

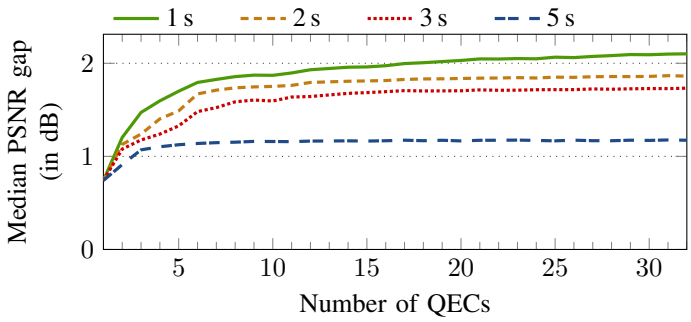

Figure 6: Median PSNR gap between the viewports of the cube map layout and the *uniEqui* depending on the number of QERs. Bit-rate: 6 Mbps

prediction from Quan et al. [17], who showed that prediction accuracy drops for time periods greater than 2 s.

C. Number of QERs

The number of QERs n represents another key trade-off. The more QERs there are, the better the coverage of the spherical video is, and thus the better the viewport quality will be due to a better match between the QEC and the viewport center. However, increasing the number of QERs also means increased storage and management requirements at the server (and a longer MPD file).

We represent in Figure 6 the median PSNR difference between the viewport extracted from the cube map layout and the same viewport extracted from the *uniEqui* layout with the same overall bit-rate budget. To modify the number of QERs, we set a number n, then we determined the position of the n QECs using the Thomson positioning problem [18]. For each head position in the dataset, we computed the distance between the viewport center and the QEC that was chosen at the beginning of the segment and we computed the viewport quality accordingly.

The best number of QERs in this configuration is between 5 and 7. The gains that are obtained for higher number of QERs are not significant enough to justify the induced storage

requirements (in particular not 30 QERs as in the Facebook system [8]). Having multiple QERs provides higher quality gains for short segments, due to the better re-synchronization between the QERs and the viewport centers. Note that a significant part of these gains stems from the cube map layout.

V. Conclusion

We have introduced in this paper viewport-adaptive streaming for navigable 360-degree videos. Our system aims at offering both interactive highs-quality service to HMD users with low management for VR providers. We studied the main system settings of our framework, and validated its relevance. We emphasize that, with current encoding techniques, the cube map projection for two seconds segment length and six QERs offers the best performance. This paper opens various research questions: (I) New adaptation algorithms should be studied for viewport navigation, especially based on head movement prediction techniques using *saliency maps* (probability of presence), extracted from the feedback of previous viewers [4]. Quan et al. [17] have recently made a first attempt in this direction. (II) New video encoding methods should be developed to perform quality-differentiated encoding for large-resolution videos. Especially, methods that allow for intra-prediction and motion vector prediction across *different* quality areas. The recent work from Hosseini and Swaminathan [5] is a first step. (III) Specific studies for *live* VR streaming and interactively-generated 360-degree videos should be performed, because the different representations can hardly be all generated on the fly.

References

[1] S. Choy, B. Wong, G. Simon, and C. Rosenberg. A hybrid edge-cloud architecture for reducing on-demand gaming latency. *Mult. Sys.*, 20(5):503–519, 2014.

[2] C.-W. Fu, L. Wan, T.-T. Wong, and C.-S. Leung. The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video. *IEEE Trans. Multimedia*, 11 (4):634–644, June 2009.

[3] V. Gaddam, H. Ngo, R. Langseth, C. Griwodz, D. Johansen, and P. Halvorsen. Tiling of Panorama Video for Interactive Virtual Cameras: Overheads and Potential Bandwidth Requirement Reduction. In *Picture Coding Symposium (PCS)*, 2015.

[4] J. Han, L. Sun, X. Hu, J. Han, and L. Shao. Spatial and temporal visual attention prediction in videos using eye movement data. *Neurocomputing*, 145:140–153, 2014.

[5] M. Hosseini and V. Swaminathan. Adaptive 360 vr video streaming: Divide and conquer! In *IEEE International Symposium on Multimedia (ISM)*, 2016.

[6] ISO/IEC 23009-1:2014. Information tech.: Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats, 2014.

[7] M. Kazhdan and H. Hoppe. Metric-aware Processing of Spherical Imagery. In *ACM SIGGRAPH Asia*, 2010.

[8] E. Kuzyakov and D. Pio. Next-generation video encoding techniques for 360 video and vr. Blogpost, January 2016. https://code.facebook.com/posts/1126354007399553.

[9] J. Le Feuvre and C. Concolato. Tiled-based Adaptive Streaming using MPEG-DASH. In *ACM MMSys*, 2016.

[10] J.-S. Lee, F. De Simone, and T. Ebrahimi. Efficient video coding based on audio-visual focus of attention. *Journal of Visual Communication and Image Representation*, pages 704–711, 2011.

[11] C. Liu, I. Bouazizi, and M. Gabbouj. Rate Adaptation for Adaptive HTTP Streaming. In *ACM MMSys*, 2011.

[12] J. D. Moss and E. R. Muth. Characteristics of head-mounted displays and their effects on simulator sickness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(3):308–319, 2011.

[13] MPEG DASH. mpeg-vr Info page. https://lists.aau.at/mailman/listinfo/mpeg-vr, 2016.

[14] K. Ng, S. Chan, and H. Shum. Data Compression and Transmission Aspects of Panoramic Videos. *IEEE Trans. Circuits Syst. Video Techn.*, 15(1):1–15, January 2005.

[15] D. Ochi, Y. Kunita, A. Kameda, A. Kojima, and S. Iwaki. Live streaming system for omnidirectional video. In *Proc. of IEEE Virtual Reality (VR)*, 2015.

[16] U. Pal and H. King. Effect of UHD high frame rates (HFR) on DVB-S2 bit error rate (BER). In *SMPTE*, 2015.

[17] F. Quan, B. Han, L. Ji, and V. Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *ACM SIGCOMM AllThingsCellular*, 2016.

[18] E. Rakhmanov, E. Saff, and Y. Zhou. Electrons on the sphere. *Series in Approximations and Decompositions*, 5:293–310, 1994.

[19] Y. Sánchez, R. Skupin, and T. Schierl. Compressed domain video processing for tile based panoramic streaming using HEVC. In *IEEE ICIP*, 2015.

[20] G. Tian and Y. Liu. Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming. In *Proc. of ACM CoNEXT*, 2012.

[21] I. Tosic and P. Frossard. Low bit-rate compression of omnidirectional images. In *Picture Coding Symposium (PCS)*, pages 1–4, 2009.

[22] VRTimes. Comparison Chart of FOV (Field of View) of VR Headsets. http://www.virtualrealitytimes.com/2015/05/24/chart-fov-field-of-view-vr-headsets/, 2015.

[23] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan. Mixing Tile Resolutions in Tiled Video: A Perceptual Quality Assessment. In *Proc. of ACM NOSSDAV*, 2014.

[24] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *ICSSC*, 2003.

[25] M. Yu, H. Lakshman, and B. Girod. A Framework to Evaluate Omnidirectional Video Coding Schemes. In *IEEE ISMAR*, 2015.

[26] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj. HEVC-Compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications. In

Proc. of ACM Multimedia, 2016.